HW

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/783,250	02/14/2001	Kallol Pal	JP920000411US1	1698

39903 7590 12/30/2005

ANTHONY ENGLAND
PO Box 5307
AUSTIN, TX 78763-5307

EXAMINER	
KENDALL, CHUCK O	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 12/30/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.**	**Applicant(s)**	
	09/783,250	PAL ET AL.	
	Examiner	**Art Unit**	
	Chuck O. Kendall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 07 October 2005.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 3, 5 – 17, 20, 22 – 30, 33 & 35 – 45 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 3, 5 – 17, 20, 22 – 30, 33 & 35 – 45 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☑ The drawing(s) filed on 2/14/01 is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is in response to amendment filed 10/07/05.

2. Claims 3, 5 – 17, 20, 22 – 30, 33 & 35 – 45 are pending.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 3,5,6, 9,13,15, 16,20,22,23, 25, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Darty USPN 6,173,440 (hereinafter Darty, [art of record]) in view of ATAC: Overview published 7/15/1998 (art of record).

Regarding claims 13, for testing a program having statements, said method comprising the steps of:

a) dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups, (Figure 3a, Darty, S102) and

b) determining the one of the groups that are executed when said program is executed while testing said program (Figure 3c, S137);

c) indicating unexecuted ones of the groups based on the ones of the groups that were determined in step b) to have been executed (14:36 – 40, see runtime (i.e. execution time) pass/fail matrix);

d) enabling a tester to executed said unexecuted groups such that said tester can ensure that all statements in said program are executed at least once (Figure 3d, S150, S153, S155, S 160 and C which flows back to B, on Figure 3c, Examiner interprets the unexecuted groups to be S148 in Figure 3d).

e) including an extra statement in each of said groups, wherein execution of such an extra enables said determining in step b) to identify an executed one of the groups corresponding to said extra statement, wherein said program is contained in a plurality of programs which in turn are contained in a class of an object oriented environment (Darty, see Figure 3c, S135 for Run TimePass/Fail, Examiner interprets identifying an executed ones to be blocks that passed);

f) enabling said tester to define a macro containing a plurality of programs lines; storing said macro in a database(5: 57- 65, see test points for macros); and

g) enabling said tester to execute said macro in the middle of testing said plurality of programs (5: 57- 65, see test points written during code execution).

Although, Darty, doesn't explicitly disclose wherein each of said groups contains a respective sequence of ones of the statements such that all the statements of such a group are executed if at least one statement of said group is executed, wherein such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed.

Darty does show the blocks of code being tested for passing and failing and upon the determination, if failed making the necessary corrections and re-executing see Figure 3d and 3c. However, the ATAC: Overview discloses on page 2, in section 3.3.1 that, " Block coverage ensures that all the basic blocks are executed at least once". Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty and The ATAC: Overview because, " a test case that executes all program statements tends to test a program more thoroughly than a test set that invokes all functions", ATAC: Overview section 3.2, 3rd paragraph.

Regarding claim 29, which is the computer program product version of claim 13, see rationale as discussed above.

Regarding claim 3, the method of claim 13, wherein said extra statements contains respective group identifiers, wherein said determining in step b) further comprises examining such a group identifier to determine a specific one of the groups which has been executed (Darty, see Figure 3b, S122 shows each test point being associated to blocks, Examiner interprets this as a means of identifying and correlating blocks).

Regarding claim 5, the method of claim 13, further comprising the steps of: grouping a sequence of the groups into a block; and

determining that said block has been executed only if all of the groups of the block are executed (The ATAC: Overview discloses on page 2, in section 3.3.1 that, " Block coverage ensures that all the basic blocks are executed at least once).

Regarding claim 6, the method of claim 5, Darty discloses all the claimed limitations as applied in claim 5. Although, Darty doesn't explicitly disclose wherein said grouping comprises: determining a language structure present in said plurality of programs as well as grouping a subset of groups present in said language structure into a block such that the statements in said language structure are presented as a block to said tester.

Darty does disclose grouping lines of code into functional blocks S102. However, the ATAC: Overview discloses language structures within several programs, see Block 1, Block 2, and Block 3, on page 3. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty and ATAC because, "...achieving completely adequate block coverage ensures that the entire program is at least executed " ATAC: overview section 3.3.1, 3rd paragraph.

Regarding claim 9, the method of claim 13, wherein said enabling comprises: enabling said tester to examine the statements associated with said unexecuted blocks such that said tester can determine arguments which would cause an unexecuted block to be executed; enabling said tester to enter said determined arguments to cause said unexecuted block to be executed (Darty, Figure 10, see process failure data and determine corrective action).

Regarding claim 15, the method of claim 13, wherein said dividing, determining, indicating and enabling are performed in a single computer system (Darty, Figure 3a, S102).

Regarding claim 16, the method of claim 13, wherein said object is generated in Java programming language (Darty, 21: 25 – 27, see Java).

Regarding claim 20, (computer program product) see claim 3 for reasoning.

Regarding claim 22, (computer program product) see claim 5 for reasoning.

Regarding claim 23, (computer program product) see claim 6 for reasoning.

Regarding claim 25, computer program product of claim 21, wherein said enabling means comprises:

second enabling means for enabling said tester to examine the statements associated with said unexecuted blocks such that said tester can determine arguments which would cause an unexecuted block to be executed (Darty, Figure 10, see diagnostics).

third enabling means for enabling said tester to enter said determined arguments to cause said unexecuted block to be executed (Darty, see Figure 3c, S135 for Run TimePass / Fail, Examiner interprets identifying an executed ones to be blocks that passed).

5. Claims 10 – 12,14, 17, 26 – 28, 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Darty USPN 6,173,440 (hereinafter Darty,[art of record]) in view of

ATAC: Overview published 7/15/1998, and further in view of Rodrigues USPN 6067639 A (art of record).

Regarding claim 10, Darty as modified by ATAC: Overview discloses all the claimed limitations as applied in claim 9 above. The combination of Darty and ATAC does not disclose wherein said argument comprises an instance of another object. Darty does disclose implementing using the Java language which does inherently have object instanciation Darty, 21: 25 – 27, see Java. However, Rodrigues in an analogous art discloses comprising instance of other objects see (Rodrigues FIG. 5, 502). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty and ATAC with Rodrigues because, object instanciation is a general practice in Object oriented languages such as C++ and Java, which enable functions and other class members to implement class objects.

Regarding claim 11, the method of claim 10, further comprises: enabling said tester to instantiate said instance of said another object (Rodrigues, FIG.5, 502); enabling said tester to assign a name to said instance, wherein said tester can enter said name to provide said instance as an argument value (Rodrigues, 13:13 – 15).

Regarding claim 12, the method of claim 11, further comprising:

receiving a string as an argument (Rodrigues,13:13 –15, see name); and determining that said string indicates that said instance is said argument value if said name matches said string (Rodrigues,13:13 – 35).

Regarding claim 14, the method of claim 13, wherein said macro is designed to examine the data structures within an instance of an object or to set the values for the variables in the object (Rodrigues, FIG., 502).

Regarding claim 17, the method of claim 13, further comprising: enabling said tester to load said class; enabling said tester to instantiate an instance of said class

(Rodrigues, FIG., 502); and enabling said tester to execute said program on said instance (Rodrigues, FIG., 504).

Regarding claim 26, (computer program product) see claim 11 for reasoning.

Regarding claim 27, (computer program product) see claim 12 for reasoning.

Regarding claim 28, (computer program product) see claim 14 for reasoning.

Regarding claim 30, (computer program product) see claim 17 for reasoning.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 7, 8, 24, are rejected under 35 U.S.C. 103(a) as being unpatentable over Darty USPN 6,173,440 (hereinafter Darty) in view of ATAC: Overview published 7/15/1998 (hereinafter "ATAC "), as applied in claim 6, and further in view of Uchihira et al. USPN 5,860,009 (hereinafter Uchihira, [art of record]).

Regarding claim 7, Darty as modified by ATAC discloses all claimed limitations as applied in claim 6 above. The combination of Darty and ATAC doesn't explicitly disclose wherein said blocks are defined hierarchically according to the inclusive relationship of language structures in said plurality of programs. However, Uchihira does disclose this feature in a similar configuration (25:55 – 60). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty and ATAC with Uchihira because, defining instructions hierarchically by different levels enables more efficient prioritization.

Regarding claim 8, Darty as modified by ATAC discloses all the claimed limitations as applied in claim 7. Although, the combination of Darty and ATAC doesn't explicitly disclose wherein said language structure comprises one of program delimiters, control structure and loop structure. Darty does disclose grouping lines of code into functional blocks S102. However, ATAC: Overview discloses language structures within several programs, see Block 1, Block 2, and Block 3, Block 1 of which includes conditional expressions on page 3. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty and ATAC because, "...achieving completely adequate block coverage ensures that the entire program is at least executed " (ATAC: overview section 3.3.1, 3rd paragraph).

Regarding claim 24, (computer program product) see claim 7 for reasoning.

8. Claim, 33, 35 – 43, 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Darty USPN 6,173,440 (hereinafter Darty) in view of ATAC: Overview published 7/15/1998, and further in view of Grey et al. USPN 6,507,842 B1n (art being made of record).

Regarding claim 42, Darty discloses a system enabling a tester to test a program having statements, said computer system comprising;

a random access memory (Figure 1, 24 see RAM);

a display unit containing a display system screen (Figure 1, 33);

an input interface (Figure 1, 28);

a processor (Figure 1, 22) dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups, (dividing said program into a plurality of groups such that every statement in the program belongs to at least one of the groups, (Figure 3a, Darty, S102 associated text);

said processor executing said program in response to instructions received from said input interface (Figure 3c, S137, and associated text);

said processor determining the ones of the groups that are executed when said program is executed (Figure 3c, S137 – S139 associated text);

said processor causing a display to be generated on said display unit said display indicating unexecuted one of the groups based on the ones of the groups that are determined to have been executed (Figure 3c, see Fail test and associated text, for displaying also see, (19:65 – 20: 10, see CAD which inherently uses a display, also see Figure 1, 33 for display);

wherein said computer system further comprises a secondary storage (Figure 1, 26), wherein said processor stores said program including said extra statement on said secondary storage , wherein said program is contained in a plurality of programs which in turn are contained in a class of an object oriented environment (19:47 – 55, see class and software design tool), wherein said processor receives a plurality of program lines representing a macro.

Although, Darty, doesn't explicitly disclose wherein each of said groups contains a respective sequence of ones of the statements such that all the statements of such a group are executed if at least one statement of said group is executed, wherein such a group is deemed to be executed if at least one of the statements of the group is executed when the program is executed.

Darty does show the blocks of code being tested for passing and failing and upon the determination, if failed making the necessary corrections and re-executing see Figure 3d and 3c. However, the ATAC: Overview discloses on page 2, in section 3.3.1 that, " Block coverage ensures that all the basic blocks are executed at least once". Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty and The ATAC: Overview because, " a test case that executes all program statements tends to test a program more thoroughly than a test set that invokes all functions", ATAC: Overview section 3.2, 3rd paragraph.

The combination of Darty and ATAC doesn't explicitly disclose said program storing said macro in a database and said processor executing said macro in response to receiving an instruction to execute said macro. However, Grey in an analogous art and similar configuration discloses in a testing environment storing in a database associated test steps and sequences (3:35 – 50). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty, ATAC and Grey because, storing the test in the database would improve reusability of the tests (see, Grey 1: 15 –18) and also make it more dynamic see (3:35 – 50).

Regarding claim 33, (a system) see claim 3 for reasoning.

Regarding claim 35, (a system) see claim 5 for reasoning.

Regarding claim 36, (a system) see claim 5 for reasoning.

Regarding claim 37, (system) see claim 7 for reasoning.

Regarding claim 38, the system of claim 34, wherein said processor receives instructions from said input interface to display the statements associated with said unexecuted blocks, said processor causing the statements to be displayed on said display unit such that said tester can determine arguments which would cause an unexecuted block to be executed (Darty, see Figure 3c, S135 for Run TimePass/Fail, Examiner interprets identifying an executed ones to be blocks that passed).

Regarding claim 39, the system of claim 38, wherein said argument comprises an instance of another object (Rodrigues, FIG.5, 502).

Regarding claim 40, (system) see reasoning in claim 11.

Regarding claim 41, (system) see reasoning in claim 12.

Regarding claim 43, (system) see reasoning in claim 14.

Regarding claim 45, the system of claim 42, wherein said input interface is connected to at least one of a mouse and a key-board (Darty, 4: 10 – 15, also see Uchihira, 12:11, note key-board and mouse devices are well known devices for use on computer system).

9. Claim, 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Darty USPN 6,173,440 (hereinafter Darty) in view of ATAC: Overview published 7/15/1998 and further in view of Grey et al. USPN 6,507,842 B1, as applied in claim 42, and further in view of Rodrigues USPN 6,067,639 A.

Regarding claim 44, Darty as modified discloses all the claimed limitations as claimed in claim 42 above. The combination of Darty, ATAC and Grey doesn't explicitly disclose said processor loads said class into said RAM in response to receiving an instruction to load said class, said processor further instantiating an instance of said class in response to receiving another instruction, said processor executing said program on said instance in response to receiving one more instruction. However Rodrigues does disclose this in an analogous art (Rodrigues,15: 37 – 40).

Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Darty, ATAC, Grey and Rodrigues because, it would enable it to be performed more efficiently in an object oriented environment.

Response to Arguments

9. Applicant's arguments filed 10/07/05 have been fully considered but they are not persuasive.

Argument (1), Applicant argues in page 12 of 17 of his response (10/07/05), that Darty doesn't teach "every statement in the program belongs to at least one of the groups".

Response (1), Examiner disagrees. In Darty FIG.3a, Darty shows Grouping lines of code into functional blocks. Applicant's plain language of claims calls for, dividing said program into a plurality of groups (Grouping lines of code) such that every statement in the program belongs to at least one of the groups (functional block). Furthermore, Applicant on page 13, 1st paragraph of his response argues that the lines of code which are grouped as taught by Darty should be interpreted as "some, but not all, lines are grouped into functional blocks", Examiner again disagrees. Darty discloses in 21:48 – 50 that, the lines of code of the computer program (*Emphasis added*) is what is being grouped into functional block and as such this would mean that all the lines in the computer program are being grouped, (also refer to FIG. 5, which shows the divisions of the program) and hence teaches Applicants claimed limitation. Furthermore Applicant in page 12, last paragraph of his response (10/07/05) points to a portion of Darty which Applicant contends as evidence to support his argument that Darty doesn't disclose his limitation of "every statement in the program belongs to at least one of the groups". Applicant recites element 102 in FIG.3A and col. 9, lines 39 – 41, which shows a progam "ABC" which he describes as only being divided into three parts. Applicant is relying on only a portion of Darty to support his argument. However, Darty in FIG.5, shows a whole program being divided into several parts, which is also supported by section 21:48 – 50, which describes that *the computer program* is grouped into blocks and not a portion of it.

Argument (2), Applicant also argues on page 13, 2nd paragraph that Darty does not teach or suggest "enabling a tester to execute said unexecuted groups such that the tester can ensure that all statements in said program are executed at least once".

Response (2), Darty in Figure 3d, shows in S153 if the Fault isolation matrix is empty (no errors/unexecuted code) then it stops, however if there exists Faults/Errors/unexecuted code the flow then proceeds to S155, it then incorporates the fix and proceeds to C' which continues to B' on Figure 3c and to A' on Figure 3b and restarts the execution process all over, hence enabling all unexecuted code to be executed at least once.

Argument (3), Applicant argues on page, 14 that Darty teaches away from the ATAC: Overview and that there is no suggestion or motivation to combine the references.

Response (3), Examiner disagrees. Applicant contends that Darty teachings the probability to the code given rise to a failure and hence would not teach that every line of code assigned to the block would be executed. Darty again in Figure 3d, shows that all the code eventually is executed at least once, following a fault matrix process which evaluates if faulty/unexecuted code exists. Although, in the previous rejection of (06/08/05) Examiner presented Darty and the ATAC OVERVIEW in combination, it would have beeen obvious in view of Darty *alone* that every line of code would have been be executed eventually as described and as addressed in response (2) above and as recited in Figure 3d of Darty.

Furthermore with regards to combining the ATAC: Overview and Darty, Examiner also believes that the 35 U.S.C. 103 rejection is made with adequate suggestion to justify the combination based on both Darty and ATAC being analogous art as well as dealing with testing and executing based on predefined criteria and evaluating decisions as true and false as identified in ATAC: Overview section 3.2. Similarly in Darty basing his testing and validation on criteria is also disclosed in Darty, see 7: 5 – 35, for reference and runtime testpoint data file.

Regarding claims 13, 29 and 42, Darty was recited to overcome the previously allowable subject matter and the reasons have been addressed above.

Regarding reapplying the Grey references as argued by Applicant on page 12 1st paragraph of his response (10/07/05), Grey's reference still discloses the claimed limitations as previously recited and applied in the last Detailed Office action of 06/08/05. Applicant has not provided any arguments to contest it as valid prior art and hence is being maintained.

Correspondence information

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chuck Kendall whose telephone number is 571-272-3698. The examiner can normally be reached on 10:00 am - 6:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TUAN DAM
SUPERVISORY PATENT EXAMINER